Parallel alignment of structured documents

Laurent Romary, Patrice Bonhomme

Chapter 11

Parallel alignment of structured documents

Laurent Romary, Patrice Bonhomme
Laboratoire Loria, France

Keywords: SGML, structured documents, multi-level alignment.

Abstract: Classical methods for parallel text alignment consider one specific level (e.g. sentences) along which two or more versions of a text are to be synchronised. This may lead to some problems when these documents are particularly long since alignment errors at some point in the text may, in the absence of any other linguistic information, propagate for some time without any chance of recovery. In this chapter we consider how multilingual parallel alignment can be based on the fact that more and more texts are now highly structured by means of tagging languages such as SGML. In particular we will describe recent efforts in multi-level alignment for which we will present the main advances as well as some of the difficulties to be dealt with, in particular when the text and its translation are associated with different encoding schemes or different encoding practices for the same scheme.

1. WHERE THE PROBLEM LIES.

The early stages of multilingual alignment systems have been contemporary to the increasing interest that the research community in computational linguistics has drawn on corpora to explore the reality of languages as they are expressed in speech or text. As a result, it has gone through the same exploratory realms as other techniques in the field, starting from very simple problems yielding generic algorithms and going on to tackle more subtle phenomena.

Indeed, the first attempts to put into correspondence a text and its translation have been based on the idea that the source and the target texts to be aligned had to be considered at the level of granularity of sentences. These were known as the main elementary units conveying autonomous meaning and as such were expected, by default, to follow a one to one correspondence rule. In this perspective, the problem of aligning two texts can be phrased as finding a synchronising path between two sequences of sentences in order to cope with those cases that do not exactly follow the one to one rule. It is thus no surprise that to do so people mainly relied on classical techniques such as dynamic time warping (DTW) which had been used for years in the speech recognition community.

At a higher level of structure, it is usually assumed that either no structural information at all is known and sentences are thus presented as one single and global sequence, or that intermediate level structures such as paragraphs do actually match between source and target text. This results in the possibility to directly align sequences of sentences corresponding to pairing paragraphs. However, even if an exact pairing may be encountered for specific documents such as official or legal texts[1], this assumption can be totally misleading for literary texts where translators indulge themselves with more flexibility regarding the local organisation of a text. In such cases one should find a way to cope with this variability without corrupting the primary text by unwanted mark-up[2]. For instance, when dealing with Plato's Republic in nearly a score of languages (Erjavec et al., 1998), it has even been observed that the translators of the French and the Romanian versions have decided upon two different organisation of their documents in terms of sections and chapters. As a whole, the dilemma that we have there is either to rely on a pre-processing stage which might prepare the text to be computed by a simple ("vanilla") aligner, or consider that each text, especially in the literary domain, has to be maintained electronically independently of any specific treatment it is to undergo.

More globally, the question that is being asked is the actual use that one wants to make of the texts that are considered for alignment. If the objective is just to keep track, within some kind of a translation memory, of the possible correspondences between the various portions of texts, it might not be necessary to implement an environment where the texts are considered for themselves and thus precisely maintained. If on the contrary, putting the texts in parallel is only

[1] This is typically the case in the Hansard corpus, or for European Union documents.

[2] This is for instance the option chosen for Multiconcord, which is strictly designed from a CALL perspective, as opposed to any document management view.

one operation out of a whole range of activities that these will be involved in, then it becomes essential to treat the text as an entity where the basic textual content is to be associated with indications — mark-up — describing it and organising its content. As a matter of fact, this is the second view that we will adopt in this chapter. It considers the text as a semi-structured document, a notion that is has been formalised recently (Abiteboul, 1997; Abiteboul et al., 1997; Buneman et al., 1996 ; Buneman, 1997), but which had been implemented for a few years in several encoding schemes for textual documents, among which SGML can now be considered as the most widely used.

The first section of this chapter will specifically present the elements which might be considered when encoding textual documents using the general framework of the TEI, focusing on those aspects which will guide the specific problem of multilingual alignment. We will then present possible computational answers to the use of structured documents for parallel text processing, starting from a generic modelling of these documents and going on to algorithmic aspects. We will take the opportunity to present prospective aspects related to mark-up semantics and sub-sentence information exploitation. The following section deals with the evaluation of these techniques, through the observation of the current implementations made in Geneva and Nancy. At last we present how the linking and pointing mechanisms available in the TEI, and from which similar representational frameworks have been derived in XML, can be used to store alignment information.

2. REPRESENTATIONAL ASPECTS: FROM STRUCTURED DOCUMENTS TO THE TEI.

2.1 A brief introduction to the TEI

The Text Encoding Initiative (Sperberg-McQueen, C.M., and Burnard, L., 1994) resulted from an initial meeting which, in 1987, put together representatives from different projects having to deal with electronic texts in the humanities. This meeting expressed the need to launch an initiative that would define guidelines allowing the academic community to share standard practices in the way they would represent and exchange documents (the so-called Poughkeepsie principles, cf. Ide and Véronis, 1995).

The TEI relied on SGML[3] as a framework for representing textual documents and thus defined a large and modular DTD[4] providing a common framework for a whole series of possible types of documents (prose, drama, dictionaries, transcription of spoken language etc.). The DTD is organised as layers where at a first level there are elements which are made available for any kind of document (the "core" tag set), then one may choose the specific "base" corresponding to the very genre to be represented. Finally, at a third level, there are optional tag sets to deal with specific phenomena such as names and dates encoding, pointing etc.

One very important consequence of the abundance of the elements provided by the TEI is that there is a non-negligible risk of encountering some variability in the way an encoder will represent a given text, all the more when two encoders work on different translations of a text. This variability first stems from the fact that a given text phenomena might not be considered in the same way by two encoders. For instance, a dialogue may be represented as such or just considered as a sequence of paragraphs. The second source of variability — which in some ways as we will see can be considered as a positive aspect — is that an encoder always has the possibility to enrich his text by adding further annotations to it. As a whole a newcomer in the TEI world might be so overwhelmed by the cornucopia of possible elements within his arm's reach that he may just experiment in directions which can eventually make the text very difficult to exploit.

As we shall see, there are different ways of coping with these two problems. On the one hand, we will have to introduce mechanisms to cope with the problem of ambiguity, or, put in another way, synonymy of elements. On the other hand, we will try to fully exploit enriched documents to improve alignment results when similar information is available in both source and target texts.

2.2 The primary structure of a text

At a first level, a TEI conformant text may be split into three parts (front, body, back) allowing one to isolate the actual content of a document, as opposed to preliminary information (e.g. forewords) or complementary information (e.g. bibliographies, table of contents etc.). As a matter of fact, such information may

[3] SGML (Standard Generalized Mark-up Language) is a standard (ISO 8879:1986) to describe mark-up languages in a formal way.

[4] Document Type Definition: describes in particular the various elements that be used to mark up a document and how these may be combine (a kind of document syntax).

either be highly dependent on the actual edition in a given language, or is prone to perturb the alignment process. Bibliographical entries for instance have been observed as one of the important source of errors in the Arcade evaluation (see Véronis and Langlais in this volume).

The next level of structure that has to be encoded in a textual document is obviously that of the main divisions it contains, either explicitly categorised as sections, chapters, etc. or corresponding to internal sub-divisions thereof. The latter may be typographically expressed by means of milestones (e.g. lines of stars) or larger paragraph separations. In the TEI, there are two ways of encoding such a structure. This can be done on the one hand by using an explicit hierarchy of numbered divisions (`div0`, `div1`, `div2` etc.) where each level is allotted a precise semantics (e.g. `div2` for chapters). On the other hand, the TEI provides a generic `<div>` element, which can be recursively used for the same purpose, considering that a 'type' attribute allows one to further categorise the actual level of division that is being considered. Consistently with what we presented in [Romary et al., 1999], we advocate the second option for both theoretical and practical reasons.

This structural level is rather consistently represented from one language to another, except in specific cases where for instance the translator has not considered a whole section of the text, or when (as mentioned for Plato), the interpretation of the exact structure of the source has lead to different textual organizations.

The last level to be considered when encoding the primary structure of a text is that of "paragraph", which, as opposed to the first two presented above, is far from having the same homogeneity. As a matter of fact, the notion of paragraph may be seen as any segment of text presenting some kind of discursive coherence and which may not be further divided. The TEI provides a general use element (`<p>`) to encode paragraphs, which is perfectly appropriate for marking up stream prose for instance. Still, once prose is interspersed with specific objects like lists of items or dialogues, it is observed that encoders usually proceed in two phases such that, in the first instance, they use `<p>`, as a possibly temporary tag for the text segment they want to identify, to further refine their judgement when choosing a more suitable element.

As a whole, the TEI offers the possibility to go towards the definition of reference materials which, once they are identified as such and consistently improved, are given the status of primary documents. These, depending on the application context, may be used for a wide variety of treatments. Still, since perfection is in no way to be achieved, it is necessary for the corresponding processes to be able to adapt to the possible flaws or variations in the encoding.

2.3 Enriching a textual document

The notion of primary document that we uphold in this paper corresponds to the encoding of the sole information that is directly and possibly unambiguously drawn out of the text itself independently of any theoretical attitude or at least annotator subjectivity. Once a reference version of a text has been encoded, the mark-up framework provided by the TEI allows one to further identify specific phenomena.

One of the major notions to be identified when aligning parallel texts is that of sentence. Classically, systems that align texts at the level of sentences do so by segmented the source and target documents on the fly. However, there are several arguments in favor of a preliminary encoding of sentences within a textual document. Firstly, sentences are a level of linguistic description that is not only useful for parallel text alignment, but obviously for a wide range of other linguistic treatments. Besides, when evaluating the performances of parallel text aligners, one may want to be able to make a clear difference between errors resulting from the sentence-segmenting phase and those from the aligner proper. Finally, and related to the preceding point, a sentence segmentation task is at the same time a very difficult one from an computational point of view and one that can be agreed upon (even if there are difficulties, see below) to consider a manual correction of the corresponding boundaries.

As a matter of fact, there are two categories of difficulties associated with the encoding of sentences within a text. The first one corresponds to the ubiquity of several punctuation signs such as the dot which, apart from being a sentence boundary marker, can occur within several types of expressions such as abbreviations, numbers etc. This is mainly a problem when one is automatically marking up a text but is easily manageable by a human annotator. On the contrary, the second difficulty is for more difficult to tackle both automatically and manually and is related to the possible superimposition of sentence boundaries with other structural information such as lists, or reported speech. In this latter case, we can see from the following excerpt taken from Carrol's Alice in Wonderland, that it is far from obvious to decide uniquely as to where to put sentence boundaries.

> *"Well!" thought Alice to herself, "after such a fall as this, I shall think no thing of tumbling down stairs! How brave they'll all think me at home! Why, I wouldn't say anything about it, even if I fell off the top of the house!" (Which was very likely true.)*

At sub-sentence level, encoding a text allows one to identify linguistic segments that may be given a specific status. Those can be foreign expressions, numbers, abbreviations, names etc. For some of them, the mark-up may not necessarily be coherent since for instance a foreign expression such as “a priori” can either be encoded as:

```
<hi rend=″italics″>a priori</hi>
```

at a low mark-up level, or:

```
<foreign lang=″lat″>a priori</foreign>
```

when refining the status of the corresponding object.

If one wants to go even deeper in the annotation of his text, there are possibilities to trace more subtle linguistic phenomena such as reference for instance (Bruneseaux and Romary, 1997). All these objects can then be sources of information for the processing of parallel texts, since they are not dependant *a priori* on any specific language.

2.4 Documenting textual resources

To draw a final picture of text mark-up in the framework of the TEI and what it may bring to the problem of parallel text processing, it is necessary to say a word here about documentation. As a matter of fact, it is impossible to contemplate building up a well organised and maintained textual database without having a means to describe each text in such a way that it may be automatically retrieved and identified in time (edition, version etc.) and space (if it is to be duplicated, for instance). The TEI has introduced an obligatory TEI-header which, prior to any encoding of the textual content proper, provides a comprehensive setting for describing the electronic document, both from a bibliotheconomic point of view (file references, document source, etc.), and from a more descriptive point of view. In particular, one can describe some aspects of the informational content of the text, such as the languages that are present in the texts or the various characters that appear within them. Without going too deep into this aspect, we can mention the fact that such a header provides a way to relate, within a given textual database, a given text with the translations that are also available in electronic format. It is also valuable to identify the annotation level of a text so that for instance one shall know whether

sentences have already been identified or whether it is necessary to use a segmenter prior to an alignment process.

2.5 Partial Conclusions

In the first part of this paper we have tried to show the different possibilities offered by a mark-up framework such as that provided by the TEI to describe, structure and enrich a textual document. In particular the point here is to go towards the definition of a textual fund which has some chances to last, since it is based on information which may have been checked and thus is dedicated to be reliable. In the following sections, we show different ways of using the information that has been encoded to derive specific computational process for parallel text alignment.

3. COMPUTATIONAL ASPECTS

3.1 Document structure — first notations

As we saw, far from being a linear representation of the text content, the organisation provided by the SGML mark-up of a document can be represented as a tree structure. In this structure, each node is labelled according to the element name and leaf nodes are either the elementary character chunks (so-called PCDATA) containing no tags or empty elements. For instance, the body of a typical TEI document may be represented as shown in figure 3.1.

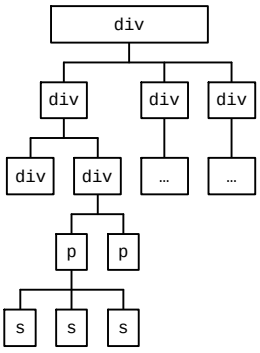


Figure 3.1: tree representation of a structured document.

This representation is very practical when one actually wants to access a given element through the expression of a path leading to it along the tree, as used for instance in the XPointer specification associated to XML[5]. Still, we will adopt in the rest of this paper a different representation based on embedded sets. This representation provides the horizontal view needed for alignment mechanisms as well as keeping the hierarchy resulting from the SGML or XML representation.

We slightly simplify our representation by supposing that the documents we deal with are "balanced", that is each leaf is at the same distance from the root node of the tree structure. For TEI documents, it means that they are homogeneous in the embedding of division, paragraph and sentence elements and that no finer-grained mark-up is considered here.

If we consider that we have an initial sequence of objects U, representing for instance the elementary text chunks that we want to deal with, we can see the document as an embedded structure represented by a series of sequences $U^i = [u_1^i, u_2^i, \ldots, u_{ni}^i]$, with the following properties:

1. $U^0 = U$, that is the initial sequence of elementary text chunks;

   For each i<n, U^{i+1} is a partition of U^i, that is

$$\bigcup_{u \in U^{i+1}} u = U^i \text{, and}$$

[5] XML (eXtended Markup Language) is a sub-set of SGML that has been defined by the World Wide Web Consortium to describe documents that will transit on the web in the future. From the point of view of this paper, the two standards (or more or less so) can be seen as equivalent. See http://www.w3c.org for further information.

$$\bigcap_{u \in U^{i+1}} u = \varnothing ;$$

2. The cardinality of the top sequence U^n is 1 (root node of the document).

In addition to this, if we suppose that we have a set I[6] of objects representing the possible element names in our document, we can label each level by means of a function L^i from U^i to I, such that for each u in U^i, L(u) is the Generic Identifier of the corresponding element. From a theoretical point of view, the ideal case is the one where each level is homogeneously marked-up with a given element, which means that for each i, L^i is a constant.

As we shall see, the actual computation of multilevel alignments is to deal with real documents where these simplifications do not fully apply. Still, this modelling of a hierarchical document seems to us a good start to formalise the generic mechanisms of multilevel alignment.

3.2 Modelling multilevel alignment

At each level, we consider that sub-trees are being put into correspondence according to a classical alignment algorithm such as those developed on the basis of a dynamic time warping (DTW) method. The corresponding result is then used as the reference framework for aligning elements at the level directly below.

To do so, we introduce the following notations:

If S and T are respectively the source and target text to be aligned, both seen as sequences of text chunks:

$S = [s_1, s_2, \ldots, s_n]$
$T = [t_1, t_2, \ldots, t_m]$

A translation alignment as produced by a procedure Align(S,T) can be described as a sequence of couples (σ_j,τ_j):

$Align(S,T) = [(\sigma_1,\tau_1), (\sigma_2,\tau_2), \ldots, (\sigma_r,\tau_r)]$

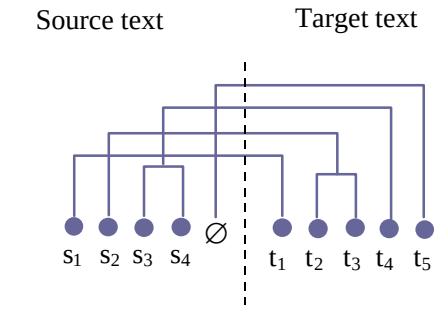
Where σ_j and τ_j are sub-sequences of S and T so that $\{\sigma_j\}_{j=1\text{-}r}$ and $\{\tau_j\}_{j=1\text{-}r}$ are respectively a partitions of S and T, that is:

$$\bigcup_{j=1-r} s_j = S, \text{ and}$$

[6] For instance, I={div,p,s …}

$$\bigcup_{j=1-r} t_j = T$$

This notation is intended to represent any kind of n to m alignment depending on the cardinality of σ_j and τ_j. For instance, 0-m or n-0 alignments are represented by couples where $\sigma_j = \varnothing$ (resp. $\tau_j = \varnothing$). A simple example is thus represented in figure 3.2, where different kinds of alignments may be observed.



Align(S,T) = {([s_1],[t_1]), ([s_2],[t_2,t_3]), ([s_3,s_4],[t_4]), (∅,[t_5])}

Figure 3.2: an alignment example.

If we now extend our notation to deal with multilevel alignments, from level 0 (bottom) to n (top), as described above, we can represent the alignment relation at level i as:

Align^i(S^i,T^i) = [(σ_1^i,τ_1^i), (σ_2^i,τ_2^i), …, (σ_{ri}^i,τ_{ri}^i)], with the same constraints on σ_1^i and τ_1^i in relation to S^i and T^i as those we had for the mono-level case, to which we can add the following coherence rules associated with the multiple levels:

Align^n(S^n,T^n) = [(σ_1^n,τ_1^n)], where $\sigma_1^n = S^n$ and $\tau_1^n = T^n$

For each i<n, if (σ_j^i,τ_j^i) belongs to Align^i(S^i,T^i), then there exists an index k such that,

3. (σ_k^{i+1},τ_k^{i+1}) belongs to Align^{i+1}(S^{i+1},T^{i+1}), and

we have both $s_j^i \subset \bigcup_{s \in s_k^{i+1}} s$, and $t_j^i \subset \bigcup_{t \in t_k^{i+1}} t$

The first condition expresses the fact that the two top nodes are automatically aligned with one another, which will correspond to the initial statement of the

algorithm. The second condition is the coherence rule by which any alignment defined at level i is necessarily a refinement of an alignment pair at level i+1.

In simple cases where we deal with one to one alignments, this condition is rather obvious. It simply says that for example when two paragraphs have been aligned, the sentences belonging to the paragraph in the source text can only be aligned with the sentences belonging to the associated paragraph in the target text, as exemplified in figure 3.3.

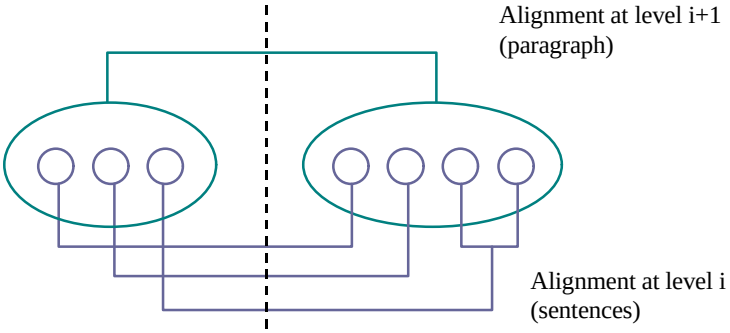


Figure 3.3: A coherent alignment at the levels of paragraphs and sentences

For more complex cases, this constraint allows elements at level i to be recombined, even if they belonged to two different units at level i+1, provided that the units have been put together (kind-of merged) within an alignment couple in $Align^{i+1}(S^{i+1},T^{i+1})$. Such a case can be seen in figure 3.4.

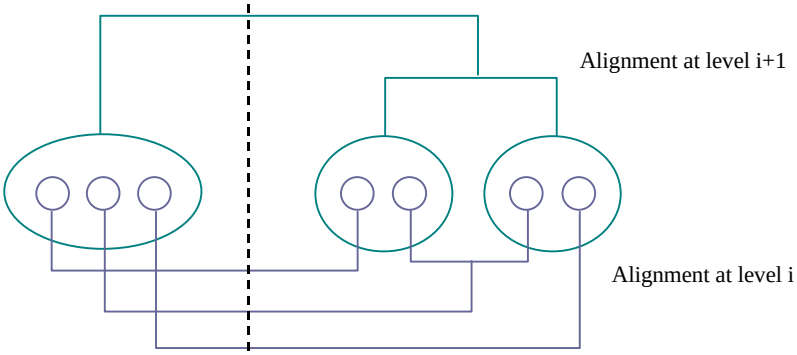


Figure 3.4: A recombination at level i compatible with an alignment at level i+1

From the specifications presented above, we can derive a generic algorithm to compute multi-level alignments. This algorithm presupposes that there exists a procedure Align(S,T) which is able to align two sequences S and T (e.g. two sequences of paragraphs) at a given level . This procedure can be based on a classical statistical method, even if we shall suggest some possible variations in sections 3.3 and 3.4.

Multi-level alignment algorithm (MLAlign):

```
MLAlign(S,T)
      /* Checks whether S and T can be compared (see 3.3) */
      If not Comparable(S,T)
          Return(ERROR)
      /* Checks whether S (hence T) contains elementary chunks
   */
      If ContainLeaves(S)
          Return({(S,T)})
      /* Chunks in the source text which have been put
   together by an alignment at the upper level are merged */
```

Let $\Sigma = \bigcup_{s \in S} s$

Let $T = \bigcup_{t \in T} t$

```
      /* We rely upon a standard procedure to align elements
   at this level */
   A-couples = Align(Σ,T)
   Res = ∅
   For each (σ,τ) in A-couples
          Res = Res ∪ MLAlign(σ,τ)
      /* The result is the set of all pairs at the finest
   grained level */
   Return(Res)
```

Several remarks can be made about this procedure:

- MLAlign is initially called with the two documents to be aligned as parameters, or any sub-trees within these, that one would want to compare;
- If one wants to derive intermediate results (like paragraph alignments), this can be easily obtained by bringing up a full hierarchical representation of the results provided at each level, instead of simply computing the union of those yielded by lower levels;
- In the same way, it might prove useful to constrain the algorithm not to go beyond a given representation level of the document, when for instance one only want to check that the source and target texts have exactly the same div structure. This is also a way to deal with finer grained mark-up (i.e. below the sentence level) which are not to be explored by the aligning process;
- As presented above, the algorithm is already fit for dealing with trees with unequal depths all along their structure.

3.3 Dealing with encoding discrepancies

As we have seen, the hierarchical algorithm presented above does take into account the fact that the source and the target text may not have been encoded in a strictly parallel way at the intermediate levels between the root element and the leaves (i.e. sentences). Still, real documents have even more reasons to exhibit encoding discrepancies. In particular, we would like to tackle here the problem of synonymy, which provides an opportunity to give some views in the domain of document (or mark-up) semantics.

As seen in section 2, there are different ways to encode the same portion of a text depending on the level of knowledge that one has, both of the DTD and of the text itself. For instance, the general use `<p>` element expressing a paragraph-like object might be further seen as a quotation (`<q>` element), a note (`<note>`) or a group of lines (`<lg>`). In some ways, these elements might be considered as synonyms, belonging to the same *class* and which an aligner should not regard as being completely different. More generally, the problem that we mention here is close to that tackled by Ide et al. (1997) when trying to represent information retrieval mechanisms for TEI encoded documents by means of a knowledge representation (KR) system. The authors' idea is to represent the kinship of elements within the TEI by defining classes between which could hold some relations such as one expressing that an instance of a given class may occur within an instance of another.

Still, the problem with such a symbolic representation is that it does not take into account that an <lg> element (a group of lines) is more likely to be aligned with a <p> (paragraph) than it is to be with a <figure> for example, even if all those elements are structurally similar, as they occur at the same level (below divisions). At the same time, such a KR based approach is valuable for checking the consistency of the two trees to be compared by the algorithm presented above, through a systematic test (expressed by the Comparable(S,T) function), which only allows the comparison of sub-trees with elements belonging to the same class. Complementary to this, we can implement a confusion matrix, which, within one given class, will express weights that can be used in the alignment process to modify the distance that is computed between two text segments (usually on the basis of the number of characters within each segment). A possible matrix is thus shown in figure 3.5 for the class corresponding to head, q, p, lg and figure elements.

	head	p	lg	figure
head	1.0	0,7	0,2	0,2
p	0,7	1,0	0,8	0,8
lg	0,2	0,8	1,0	0,3
figure	0,2	0,8	0,3	1,0

Figure 3.5: a confusion matrix between elements of the same class

3.4 From cognates to cogtags

There is a further improvement that can be brought to the algorithm and which exploits the fact that texts may be encoded at a deep level. Indeed, if we consider sub-sentence mark-up that identifies meaningful text sequences such as names, numbers, foreign expressions or abbreviations, these can be used as anchors to guide the alignment process. As a matter of fact, when dealing with a pair of texts, whether they have been encoded independently or not, there is some chance that similar phenomena have been identified and marked-up[7]. Hence, what we contemplate here can be related to the notion of cognates, which has been widely used in recent years to improve the accuracy of aligners. Cognates correspond to linguistic sequences that are morphologically similar from one language to another and are thus likely to express a similar content

[7] A specific field in the TEI header (tagUsage) is dedicated to the description of the set and number of tags used in the marked up text and can be thus used to derive the potentiality of the two text to be compared at sub-sentence level.

when encountered in the source and target texts to be aligned. Even if on some occasions the morphology of two languages may be related enough to provide pairs within their core lexicons, the best candidates are usually more specific elements such as proper names or numbers. It is easy to extend this notion in the domain of structured documents with even more accuracy since, when such a notion as a name has been tagged, there is a clear-cut decision to be taken if one wants to associate the corresponding sentences.

3.5 Evaluation

3.5.1 General observations

There are currently two systems that are sensitive to the logical structure of multilingual resources: the system implemented at LORIA (whose general strategy is presented in this chapter), henceforth MLAlign, and the one implemented at ISSCO. The latter, as opposed to the former, is based upon a two steps algorithm which, in the first instance, corrects structural discrepancie, so that the second phase can rely on a strict parallel structure at higher levels to compute correspondences between sentences. Still, both systems share the same interesting property that, when an error occurs at some point in the text, there is no propagation of it beyond the current structural context (e.g. the paragraph when aligning sentences). Unlike a classical alignment procedure, a multi-level aligner does not need a specific mechanism to synchronise the current alignment to avoid the propagation of errors until the end of aligning process.

To provide another criterion of comparison, we can say a word here about the respective perplexity of the "classical" alignment algorithm as compared to the hierarchical one, as described in section 3.2 above. To evaluate this, we can consider that we have to align 2^n sentences in two languages. A DTW-like algorithm will have to explore, independently of any specific heuristic that might be added to the basic process, a table that has a size close to 2^nx2^n. The perplexity of the algorithm is thus close to 2^{2n}. As compared to this, we can approximate the hierarchical algorithm by considering that it operates on a binary tree of depth n, and has to explore, at each level i, 2^{i-1} tables of size 2x2. The total sum of the number of operation is thus: $\sum_{i=1}^{n} 2^2 \times 2^{i-1}$, which equals to 2^{n+2}-4. The gross perplexity of a hierarchical algorithm is thus of the order 2^n, which is the square root of the classical method.

3.5.2 Evaluation within the ARCADE campaign

The ARCADE campaing allowed us to evaluate and compare the *MLAlign* with other alignment systems. The conclusions of the evaluation are the following ones:

– The MLAligner is (together with the ISSCO system) the only one that has based the dynamic alignment on the logical structure of resources;
– The quality of an alignment (in term of precision and recall rates) are highly correlated to the quality of the resource encoding.

It is interesting to notice that when a text had been very poorly encoded, such as was the case with Jules Verne's *De la terre à la lune* for which even paragraph structure had been lost, MLAlign will rate very low. On the contrary, it ranks first when structure has been preserved, which is highly promising if one considers that MLALign did not use any linguistic information at all.

4. USING SGML/XML MARK-UP TO ENCODE ALIGNMENT RESULTS

Seeing that we do not want to adapt each text for a specific processing, encoding alignments is a very important task. We tried not to clutter the text content itself with another set of tags for marking up the whole set of alignments to the target texts. Encoding alignments is also connected to the intention of the user. If his intention is to handle and store a large set of multilingual and aligned texts, it is necessary to use a well-established linking mechanism for handling multiple internal and external links between a source text and its target texts. In this section we give a quick overview of how the TEI can be used to represent and store multilingual alignments, keeping in mind that the approach taken here is complementary to that followed when trying to represent translation memories and developed by Gerhard Budin in this book.

There are several constraints that have led us to use a TEI encoding mechanism to represent parallel texts. First, as expressed earlier in this chapter, we want to manage multilingual resources independently of their specific use. In this context, there is a strong necessity not to duplicate textual content since it may induce some difficulties to update information when needed. Moreover, we had in mind that it might be possible to made some slight corrections to the texts

(typos, sub-sentence annotation improvements etc.), without interfering with the available alignments. On the side of the alignments proper, we wanted to have a way to edit them manually and complementary to that to be able to maintain alignments derived from different processes and possibly at different levels.

The TEI guidelines provide four different combinations for encoding alignments:

- two possibilities to identify the aligned portion either with points between text chunks (`<anchor>` tags) or with identified segments (`<seg id="seg1">…</seg>, <s id="s1">…</s>`);
- two possibilities to encode the alignments either with cross-references (`ID/IDREF`) using the attribute `corresp` within the `anchor` tags or by gathering a whole set of alignments within a `linkGrp` element (group of `link` elements) at the end of the source text.

According to the preceding constraints, we made the choice to use the identified segment of text chunks with `linkGrp` for alignment encoding. The following example shows a possible implementation of this solution. The `linkGrp` element puts together three different types of information. It first identifies the segments of the target texts that will be aligned by means of external pointers. It then links together sequences of textual segments which are being put together in case of multiple alignments (e.g. 2-1, 1-2, 2-2 etc.), which we call vertical links. The final series of links represents the alignments proper, as it relates objects in the source and target texts.

```
<linkGrp domains="b1 b1" targType="s" targFunc="FR EN"
targOrder="Y" evaluate="all" crdate="29 Apr 1999"
doc="LePetitPrince-EN.sgml" type="alignment">
   <xptr id="x1" from="id(d1p1s1)">
   …
   <xptr id="x143" from="id(d3p10s4)">
   …
   <link id="l9" type="linking" targets="d2p16s6 d2p16s7"/>
   …
   <link id="l18" type="linking" targets="x143 x144"/>
   …
   <link targets="d1p1s1 x1"/>
   <link targets="l9 x105"/>
   <link targets="d3p8s1 l18"/>
```

```
   …
<linkGrp/>
```

The multi-level alignment system can be used with some other types of resources in terms of encoding schema. In that case, it could be important to consider a more generic mechanism for encoding the alignments. We propose to adapt the XML pointers and linking system for our purposes.

Inspired from the TEI extended pointers, the XML pointers are based on the same functionalities. In particular, we are only using the absolute location terms (ID location) of the XML pointers. The following example thus shws an XML compliant version of our alignment representation schema.

```
<linkGrp xml:link="group"
            …
            doc="http://www…/toto.fr#">
   <xptr id="x1" href="id(s1)">
   …
   <link targets="s1 x1"/>
   …
<linkGrp/>
```

In the long run, it will be necessary to fully externalize alignment information, which might point to resources independently of their actual location. We expect that linguistic resource management will be more and more decentralized in the future and thus that appropriate representational means should be devised and implemented.

5. CONCLUSION

In this chapter, we have tried to provide a wide picture of the possible consequences resulting from having structured linguistic resources (in particular SGML or XML encoded texts) at our disposal in the specific context of parallel text processing. We have seen that this could be considered both at the algorithmic level and at the representational level. As we expect that more and more texts will be made available as structured documents within large repository funds, some of the techniques presented here will probably make even more progress in the future. Still, the methodology that we have presented is not

limited to parallel text processing, since it corresponds to a larger view of text as an organised structure beyond the linguistic content.

6. REFERENCES

Abiteboul Serge (1997), Querying semi-structured data. *Proceedings of ICDT*, Jan 1997.

Abiteboul Serge, Dallan Quass, Jason McHugh, Jennifer Widom, & Janet L. Weiner (1997). The lorel query language for semi-structured data. In *Journal of Digital Libraries*, 1(1), 1997. (cf. http://www-db.stanford.edu/pub/papers/).

Bruneseaux Florence & Laurent Romary (1997), Codage des références et coréférences dans les dialogues homme-machine, *Proceedings of Joint International Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing*, Kingston (Ontario).

Buneman Peter, Suzan Davidson, Gerd Hillebrand, & Dan Suciu (1996) A query language and optimization techniques for unstructured data. *Proceedings of ACM-SIGMOD International Conference on Management of Data*, pp. 505-516, Montreal, Canada, juin 1996.

Buneman, P. (1997). Semistructured data, Tutorial presented at PODS'97.

Erjavec, T., Lawson, A., & Romary, L. (1998). East meets West: Producing Multilingual Resources in a European Context. *First International Language Resources and Evaluation Conference*, Granada, Spain.

Ide, Nancy, and McGraw, Tim, & Welty, Chris (1997). Representing TEI Documents in the CLASSIC Knowledge Representation System. *Proceedings of the Tenth workshop of the Text-Encoding Initiative*. November, 1997.

N. Ide. & J. Véronis, eds. (1995). *The Text Encoding Initiative: Background and Contexts*, special triple issue of *Computers and the Humanities*, 29(3).

Romary Laurent, Patrice Bonhomme, Florence Bruneseaux & Jean-Marie Pierrel (1999). Silfide: A System for Open Access and Distributed Delivery of TEI Encoded Documents, *Computers and Humanities*, 33(1-2), 31-38.

Sperberg-McQueen, C.M., & Burnard, L. eds. (1994). *Guidelines for Electronic Text Encoding and Interchange*. Chicago and Oxford.